

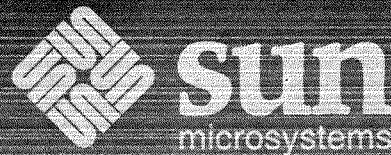
Edy Harvey

HOTJAVA

Bringing Interactivity to the Internet

Dr. Eric Schmidt

Sun Microsystems, Inc.





What is HotJava?

- ◆ Less than Mosaic: it understands no protocols or data types
- ◆ More than Mosaic: it knows how to find out about things it doesn't understand
- ◆ Based on the ability to safely download code
- ◆ From remote hosts



Background of Internet

- ◆ Began in 1950's as government/education "indestructible" nationwide network
- ◆ 1990's:
 - government role diminished
 - Mosaic launched the Web revolution
- ◆ Today:
 - 30+ million users
 - 4.8 million hosts
 - 27,000 Web servers, doubling every two months
 - 1 million active Web users



The Web Today

- ◆ An eclectic environment
 - vast series of interconnected networks
 - wide range of platforms
 - constantly evolving protocols
- ◆ Security not designed in
- ◆ Bandwidth strained to limits
- ◆ No real interactivity

*New focus on commerce, but expectations
are beyond current capabilities*



Conventional Web Browsers

- ◆ Limited to static viewing of text and images
- ◆ Pages can't contain behavior
- ◆ Can't handle new, unfamiliar protocols
- ◆ Platform dependent
- ◆ Clients are "dumb"



HotJava: *Dynamic* Web Browser

- ◆ Brings true interactivity to the Web
- ◆ Transforms static Web pages
- ◆ Embeds live applications
- ◆ Extensible: “learns” new protocols on the fly
- ◆ Fully compatible with existing browsers
- ◆ No added system/bandwidth impact



HotJava: Comprehensive Solution

Based on Java object-oriented language

- ◆ Dynamic
 - applications load transparently
 - applications are extensible
 - no software installation necessary
- ◆ Secure
 - maximum virus protection
 - encryption and authentication
- ◆ Architecture-Neutral
 - runs on any OS and CPU
- ◆ Distributed
 - local and remote access to objects



Java: Programming for the Net

- ◆ Network-Centric Application Development
 - easy to learn and use
 - reduces bugs
 - secure
 - extremely scalable
 - dynamic
 - high performance
 - cross-platform
 - rich class libraries for GUI
 - network-ready
- ◆ Create Interactive, Secure, Distributed Apps



HotJava Brings the Web to Life

Enabling:

- ◆ real-time portfolio management
- ◆ interactive advertising and shopping
- ◆ customized, animated newspapers
- ◆ live spreadsheets
- ◆ interactive advertisements
- ◆ 3-D science simulations
- ◆ live chats, online games
- ◆ interactive customized forms
- ◆ ...and much more

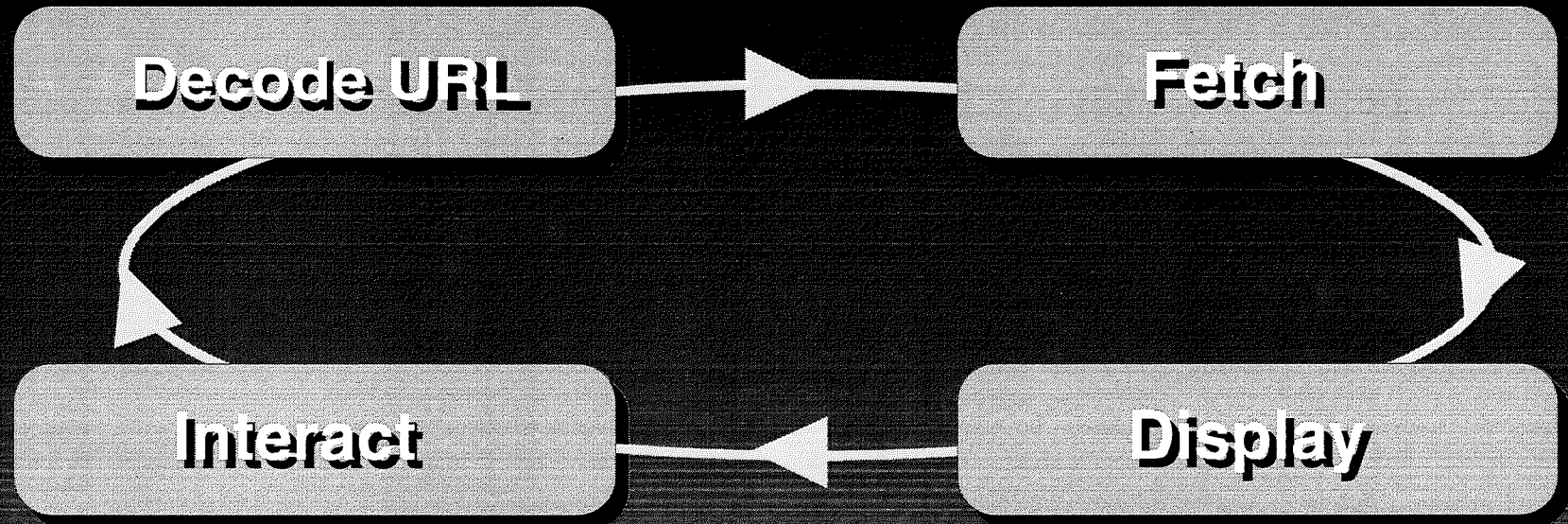
Project History

- ◆ Research looking into building:
 - Small, reliable, safe, long lived, processor independent, real-time, distributed systems
- ◆ C++ started to crumble:
 - Processor independent & long lived required different compiler technology
 - When we looked into reliable and safe, the language had to change too
- ◆ Then the WWW happened!



WWW Background

- ◆ Mosaic: a browser for the WWW
- ◆ The basic structure is simple



- ◆ Significant complexity from the many data types and protocols supported



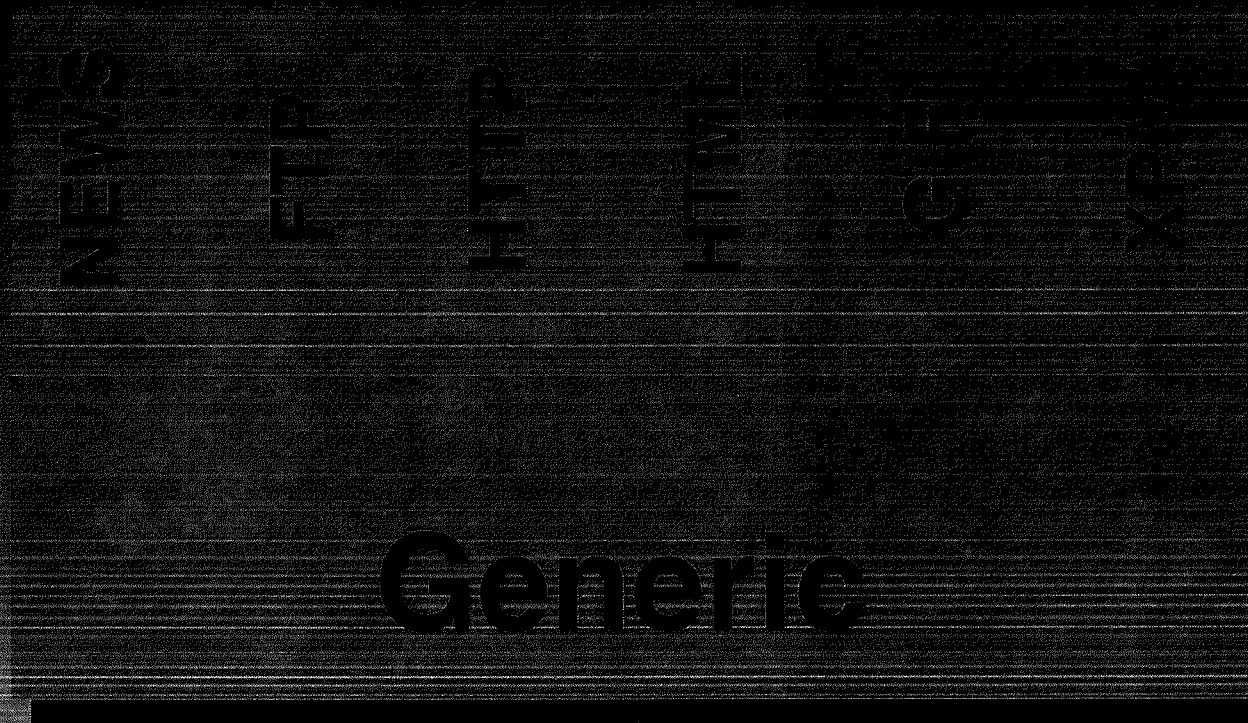
The WEB & Mosaic

- ◆ Pages are essentially static things.
 - Very book-like
 - With amazing indexing
- ◆ Depends on locally installed handlers
- ◆ New behavior requires new standards and readers
- ◆ Getting new things out is hard



Conventional Browser

- ◆ Many standards built in



- ◆ It's a solid unchanging chunk



HotJava...

- ◆ Just a platform for plugging things in:

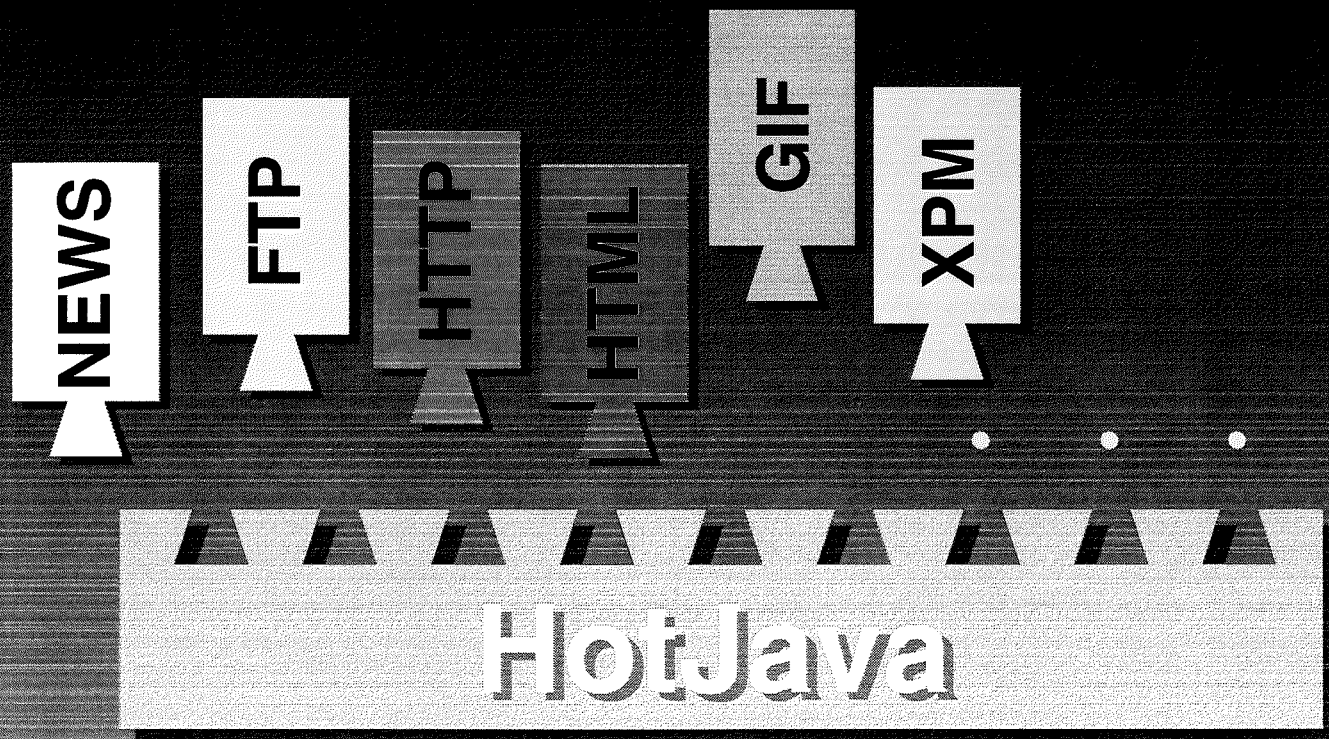
The HotJava logo, which consists of a horizontal rectangular bar with a series of small, dark, triangular shapes along its top edge. The word "HotJava" is written in a bold, sans-serif font across the center of the bar.

HotJava



HotJava...

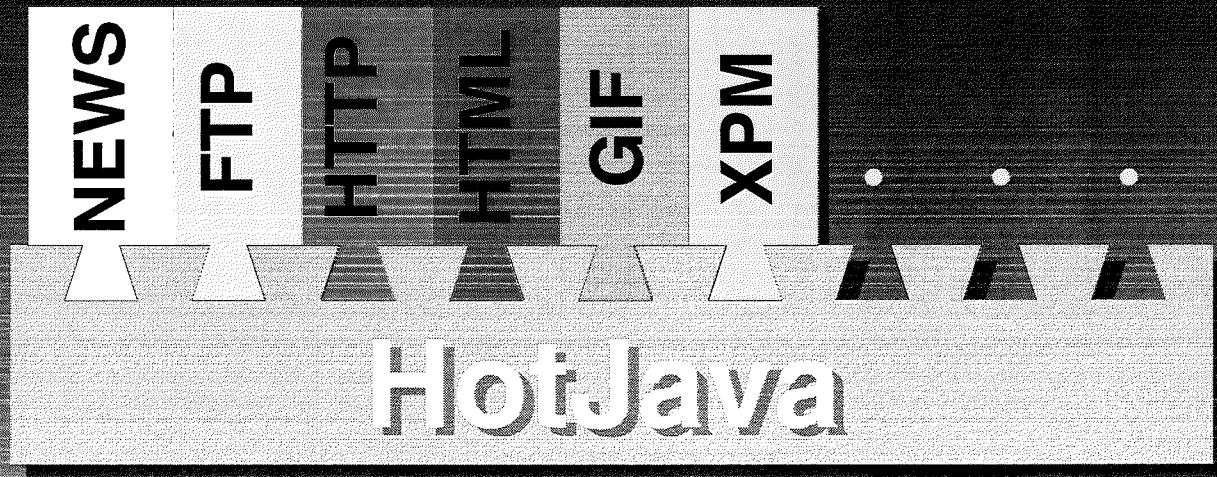
- ◆ Implementations of standards come in from the outside:





HotJava...

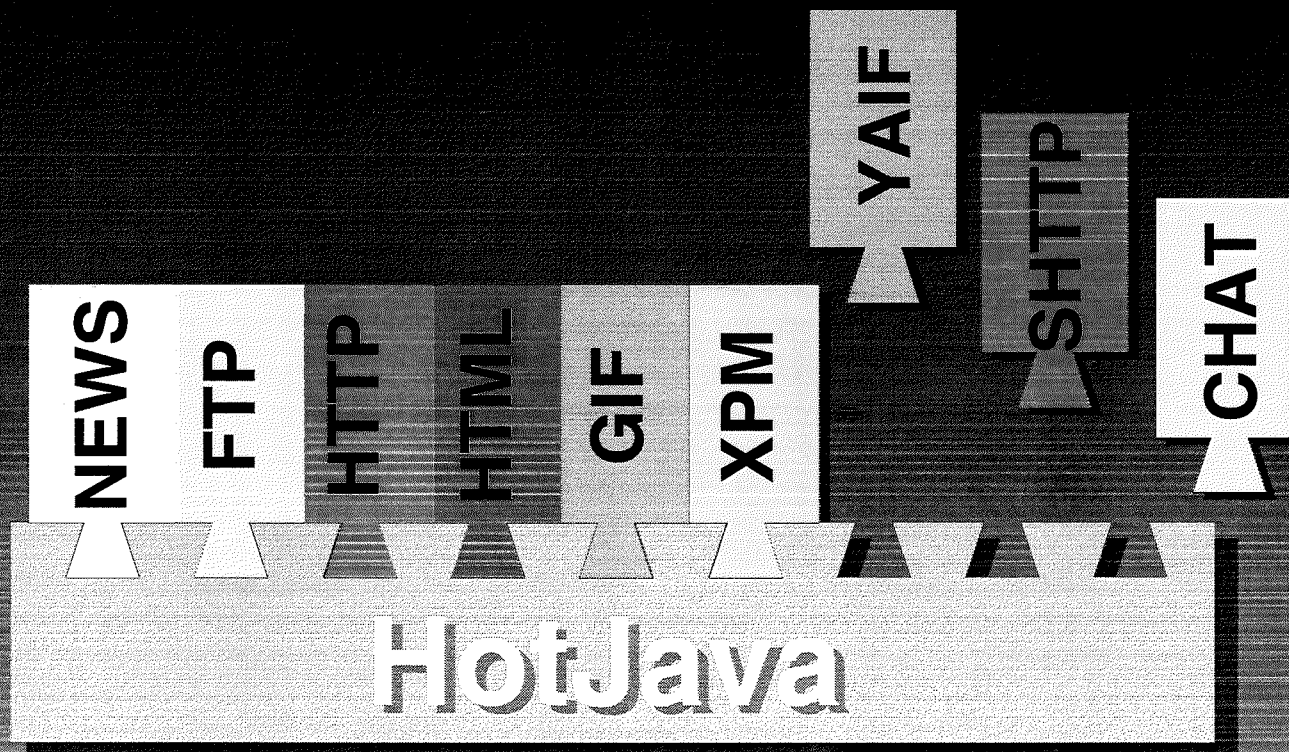
◆ And snap into place:





HotJava...

- ◆ Can have new, unthought-of things brought in later:





What can you do with it?

- ◆ New protocols: if it doesn't understand the protocol field of a url, it finds out
- ◆ New data types: if it doesn't understand a mime type field, it finds out
- ◆ New embedded items: if it finds one it doesn't understand, it finds out



Applications

- ◆ html pages with new kinds of items (eg. a fractal curve)
- ◆ html pages with interactive items (eg. embed in a physics text an interactive simulation)
- ◆ invent new protocols without having to install new clients all over the network
- ◆ have many protocols (eg. security and billing) from several sources work together in the same client